INTERNET-DRAFT
<draft-kitamura-ipv6-record-route-00.txt>
Expires in six months

H. Kitamura
NEC Corporation

17 November 2000

Record Route for IPv6 (RR6) Hop-by-Hop Option Extension

<draft-kitamura-ipv6-record-route-00.txt>

Status of this Memo

Abstract

This document proposes a "Record Route for IPv6 (RR6)" mechanism. It is composed of one new IPv6 hop-by-hop option (RR6 option) extension.

The RR6 mechanism is redesigned to establish an optimized record route mechanism for IPv6. Two types of new features are introduced.

1. In order to prevent scalability problems and to make the mechanism flexible, the beginning and ending points of the data recording range are controlled by using their specifiers.

2. In order to record long IPv6 addresses efficiently, a simple address compression method is introduced. Since the compression is achieved by utilizing characteristics of recorded IPv6 addresses effectively, the mechanism is efficient and optimized for IPv6.

1. Introduction

This document proposes a "Record Route for IPv6 (RR6)" mechanism. It is composed of one new IPv6 Hop-by-Hop option (RR6 option) extension.

In the IPv6 specifications [IPv6], no record route function is defined, and the record route function for IPv4 is less flexible and has scalability problems.

The RR6 mechanism is redesigned to establish an optimized record route mechanism for IPv6 and fix the problems of the record route function for IPv4. Two types of new features are introduced to achieve this goal.

1. In order to prevent scalability problems and to make the mechanism flexible, the beginning and ending points of the data recording range are controlled by using their specifiers.

2. In order to record long IPv6 addresses efficiently, a simple address compression method is introduced. Since the compression is achieved by utilizing characteristics of recorded IPv6 addresses effectively, the mechanism is efficient and optimized for IPv6.

2. Design of the Record Route for IPv6 (RR6) Mechanism

Since a record route operation runs on each node along a communication path, it is appropriate to achieve the record route operation as a hop-by-hop option operation. One IPv6 hop-by-hop option (RR6 option) is introduced to establish the RR6 mechanism.

2.1 Roles of the RR6 Option Operations

The RR6 option has the following roles.

1. Provide Data Storing Space

   A Data Space is prepared in the RR6 option header. All of the data recorded by the RR6 option operation is written into the Data Space in the RR6 option header.

   Changing the length of the packet on an intermediate node makes the mechanism complex, so that the length of the Data Space is not changed by the RR6 address data recording operations on each node. When the packet in which the RR6 option is set is prepared for sending, the Data Space is preallocated in it.

(Since the length of the Data Space is limited, the data recording or carrying capacity of one packet in which the RR6 option is set is limited. This scalability issue is discussed in Section 2.2)

2. Record Address Information

When a packet in which the RR6 option is set passes through nodes along the communication path, the RR6 option operation routine on each node records (incoming interface's) address information of the node. The address information is written into the Data Space in the RR6 option header.

Exception:
On the source node, outgoing interface's address information of the source node is recorded.

3. Carry the Recorded Data

The recorded data is carried by packets in which the RR6 option is set. This is easy to achieve, because the Data Space exists in the RR6 option header.

(Since recorded data is normally analyzed on a source node that issues the packet in which RR6 option is set, the recorded data must be carried back to the source node. Namely, the RR6 option must be set to the packet that returns to the source node. This issue is discussed in Section 2.4)

2.2 Data Recording Range Control

Since the length of the Data Space is limited, the data recording or carrying capacity of one packet in which the RR6 option is set is limited.

To prevent scalability problems and to make the mechanism flexible, the beginning and ending points (nodes) of data recording range are controlled by using their specifiers.

When a packet in which the RR6 option is set can not record or carry all of address information of nodes along the communication path (because the number of nodes is larger than the capacity), an additional packet(s) whose data recording beginning point specifier is set appropriately is issued and the remaining address information is recorded and carried by it.

This RR6 mechanism does not cause any scalability problems. Details of the data recording range control are explained in Section 3.3.

2.3 Address Recording with Compression

The maximum length of one hop-by-hop option is 255 octets, and the IPv6 address is long in length (16 octets). If address information is recorded as a plain (non-compressed) address format, one hop-by-hop option can record and carry only 15 record entries.

When the RR6 mechanism is applied to a large-scale network environment, 15 entries are insufficient. Thus, it is necessary to provide a compressed address recording method.

It is requested that a compressed address recording method of the RR6 mechanism must be sufficiently simple and efficient, because the RR6 mechanism is achieved as a hop-by-hop option operation that runs on each node along the communication path. A complex and heavy operation is not suitable for the compressed address recording method.

A series of recorded addresses is composed of successive neighbor addresses, and is not a series of random numbers. It is strongly assumed that each address is correlated with its previous or succeeding address in the series. In other words, two neighboring addresses will have the same leading part (prefix). This special characteristic can be used for a simple and efficient address compression method.

Generally speaking, compression methods are categorized into two types, intra-compression and inter-compression.

Since inter-compression methods work efficiently for a series of successive neighbor addresses, the RR6 mechanism adopts one of the inter-compression methods to achieve a compressed address recording mechanism. Details are explained in Section 3.3,

2.4 Loop Communication Path

Since recorded data is normally analyzed on a source node that issues a packet in which the RR6 option is set, the recorded data must be carried back to the source node. Namely, the RR6 option must be set to the packet that returns to the source node.

There are two ways of making a loop communication path.

One is to utilize an ICMPv6 Echo Request/Reply (ping6) mechanism.

> In order to run the RR6 mechanism appropriately on the ping6 mechanism, the following two operations must be implemented on the ping6's target node.

1. All of the RR6 option fields must be copied from an Echo Request message to an Echo Reply message.

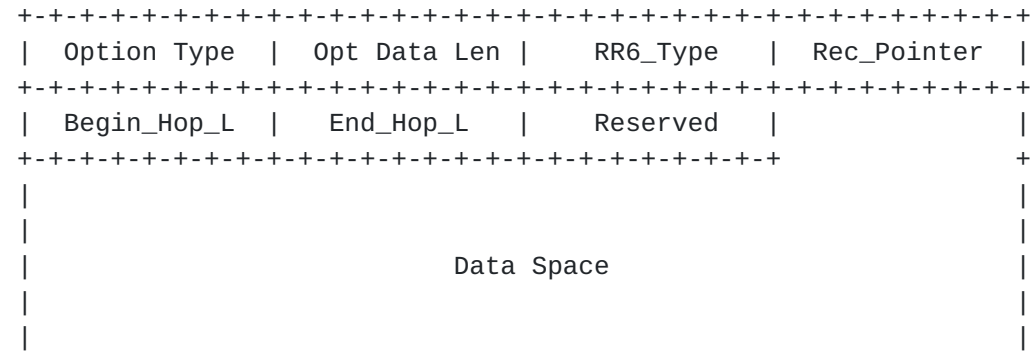
2. The Hop Limit field of the IPv6 header of these messages must be copied as a continuous sequence.

The other is to prepare a packet in which a Route option is set.

The source node must be set to the final destination of the Route option, and the target node must be set to one of the intermediate nodes that are specified in the Route option.

No special operations are required to run the RR6 mechanism in this method.

3. Record Route for IPv6 (RR6) Option Format: IPv6 Hop-by-Hop Option



Option Type: 8-bit.
Identifier of the type of option (RR6 option)
Value is 001xxxxx. 0x20+TBA
(Tentatively 0x23)

RR6 Option Type must be started as 00 to avoid discarding the packet at the RR6 feature disabled node, and the third bit must be set to 1 to record data into the [Data Space].
[see Section 4.2 of RFC2460]

Opt Data Len: 8-bit unsigned integer.
Length of the [Option Data] field of this option
Max: 255
Depends on the length of [Data Space]

RR6_Type: 8-bit.
Identifier of the data recording type

=0 Plain (non-compressed) address record type
[see Section 3.2]
=1 Compressed address record type
[see Section 3.3]

Rec_Pointer: 8-bit unsigned integer.
Pointer that shows next data recording point
in the [Data Space] in octet

Initial: 0
Data recording status can be checked by Rec_Pointer
(=0 means that no data has been recorded yet)

Begin_Hop_L: 8-bit unsigned integer.
Data recording beginning point specifier
(In a model case, it shows Hop Limit value of
the node that begins the data recording on the path)

default: 255
[see Section 3.1]

End_Hop_L: 8-bit unsigned integer.
Data recording ending point specifier
(In a model case, it shows Hop Limit value of
the node that ends the data recording on the path)

default: 0

End_Hop_L has another role to indicate that
[Data Space] is filled on the communication path.

When it is detected that [Data Space] is fully filled
and there is no space left to record new data,
[End_Hop_L] is set with the Hop Limit value of the
last data record node.
[see Section 3.1]

Reserved: 8-bit. (Reserved Octet)
Reserved octet for future use.

Data Space: [Opt Data Len]-5 octets
Buffer space to record the address information data.

Length of [Data Space] must be longer than or
equal to 17 (length of one fixed length record).

3.1 Procedure of Data Recording Range Control

As described in Section 2.2, the data recording range can be controlled by specifying the data recording beginning and ending points. Begin_Hop_L and End_Hop_L are used to specify these points.

When a packet the RR6 option is set passes through a node, the current Hop Limit value (Hop_L) of the node is initially compared with Begin_Hop_L and End_Hop_L. If Hop_L is located between them, an address recording operation is started.

End_Hop_L has another role. If the Data Space is fully filled on an intermediate node on the communication path, End_Hop_L is set with the Hop Limit value of the last data record node to indicate its location.

The following describes an algorithm of data recording range control.

```
while (Begin_Hop_L >= Hop_L && Hop_L >= End_Hop_L) {
  Prepare a data record to write into the Data Space;
  // Check the remaining space in the Data Space
  if (there is enough space to record address information) {
    Record the prepared data record into the Data Space;
    if (Data Space has been fully filled without odd remains) {
        End_Hop_L is set by the Hop Limit value of the current node;
    }
  } else { // there is no space left to record new data
     End_Hop_L is set with the Hop Limit value of
     the last (previous) data record node;
     // End_Hop_L = Prev_Rec_HopL;
     // in the compressed address record type [see Section 3.3]
  }
}
```

3.1.1 Example of Data Recording Range Control

Examples of data recording range control behavior are shown here.

```
INI: Hop Limit value of the source node that issues a packet
                [Source]-[Node1]-[Node2]-[Node3]-[Node4]-[Node5]-...
Hop Limit value     INI   INI-1   INI-2   INI-3   INI-4   INI-5  ...
(ex. INI=64)         64      63      62      61      60      59  ...
default (w/o Cont.)   x       x       x       x       x       x  ...
(Begin=(64=INI))
w/ Record Range Cont. -       -       -       x       x       x  ...
(ex. Begin=61)
w/ Record Range Cont. -       -       x       x       x       -  ...
(ex. Begin=62, End=60)
```

3.2 Plain (Non-Compressed) Address Record Type

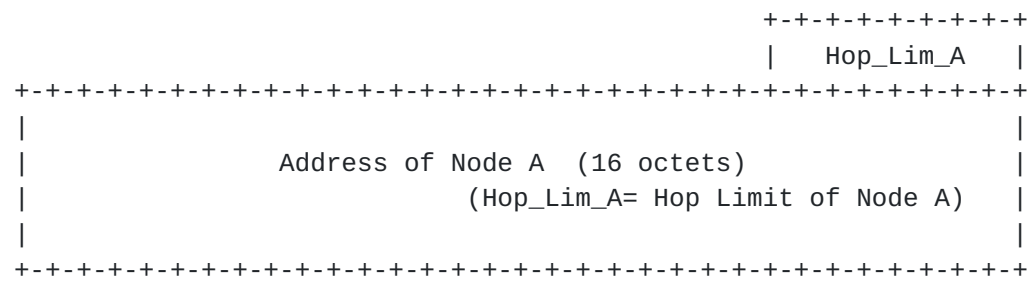
In this section, a plain (non-compressed) address record type is explained.

When [RR6_Type] = 0, a plain (non-compressed) address record type is used to record address information.

This record type is the simplest method to record address information into the [Data Space]. Address information of a node is recorded as plain format (without compression).

Each node provides one data record that contains address information. The length of the data record is fixed (17 octets). When a packet in which the RR6 option is set passes through a node, one data record space is consumed in the [Data Space].

The following is the format of one fixed-length data record.



One fixed-length data record comprises the Hop Limit (1 octet) and the (incoming interface's) address (16 octets) information of the node.

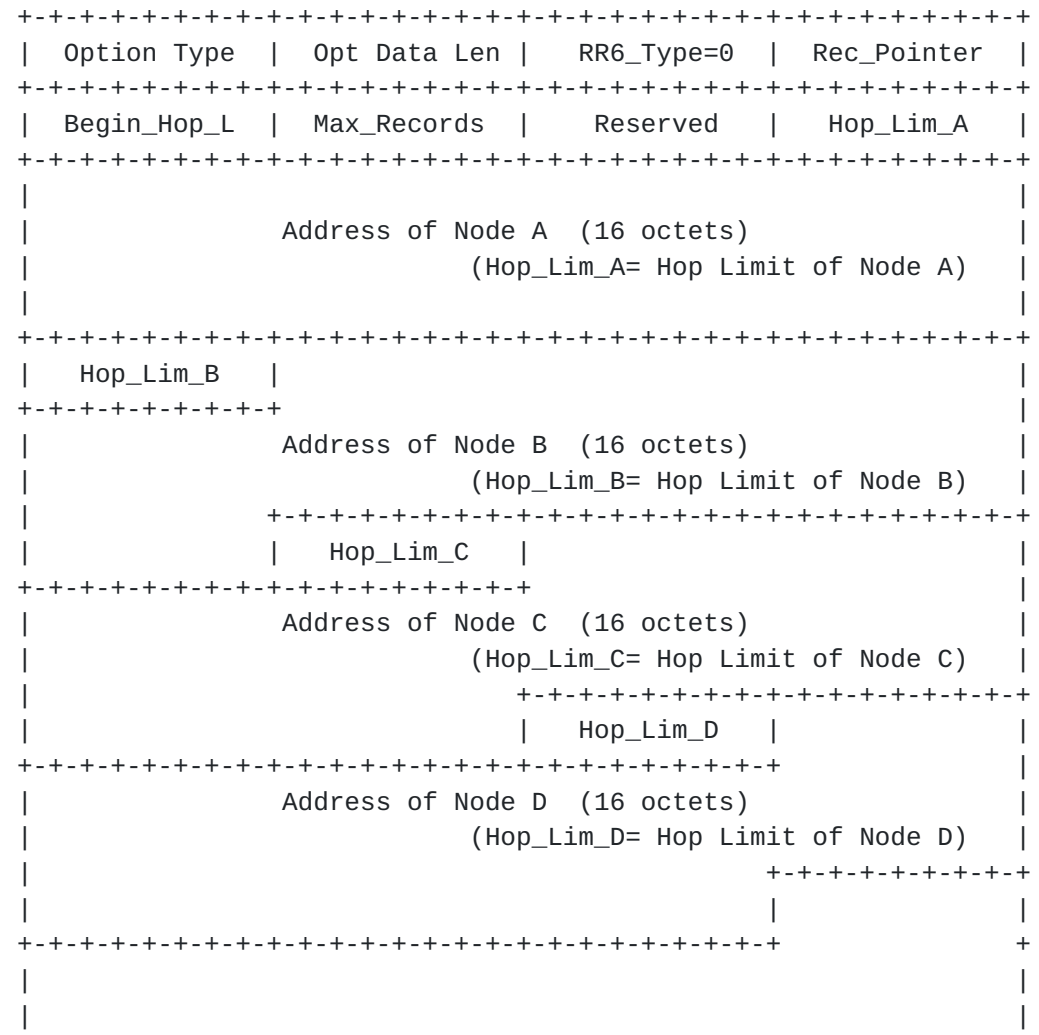
Hop Limit information of the node is used to identify the location on the communication path.

Exception:

If [Begin_Hop_L] >= Hop Limit value of the source node, the source node must record its address information. Since there is no incoming operation on the source node, outgoing interface's address information is recorded into the [Address of Node] field instead of incoming interface's address information.

3.2.1 Example of Data-Filled Format (Plain Address Record Type)

The following shows an example of how the [Data Space] is filled with fixed-length data records.



Some nodes on the communication path do not record address information for some reason (e.g., not implementing or intentionally disabling the RR6 feature).

Thus, Node A, B, C, D, ... are not always successive neighbor nodes, and Hop_Lim_A, Hop_Lim_B, Hop_Lim_C, Hop_Lim_D, ... are not always consecutive descending numbers.

3.3 Compressed Address Record Type

In this section, a compressed address record type is explained.

(This document defines one simple and efficient compressed address record type. Since there are various address compression methods, other compressed address record types may be proposed in other documents in future).

When [RR6_Type] = 1, a compressed address record type is used to record address information.

Address information of a node is recorded into the [Data Space] as a compressed format. Each node provides one data record. Since address information is compressed, the length of the record is variable (max 17 octets). When the packet in which the RR6 option is set passes through a node, its RR6 operation consumes one variable-length data record space in the [Data Space].

3.3.1 Compression Method

Series of recorded addresses have a special characteristic. It is composed of successive neighbor addresses. It is strongly assumed that each address is correlated with its previous or succeeding address in the series. Specifically, it is assumed that two neighboring addresses have the same leading part (prefix).

By omitting the same leading part, address information can be compressed. This simple compression method utilizes the special characteristic efficiently, and it is adopted for this record type.

(This method is categorized into inter-compression methods. In order to simplify the mechanism, intra-compression methods are not used in this record type.)

To establish a simplified compression mechanism, the same leading part is omitted in octet unit (not bit unit).

The compression is composed of the following operations.

1. Compare current address with previous address.

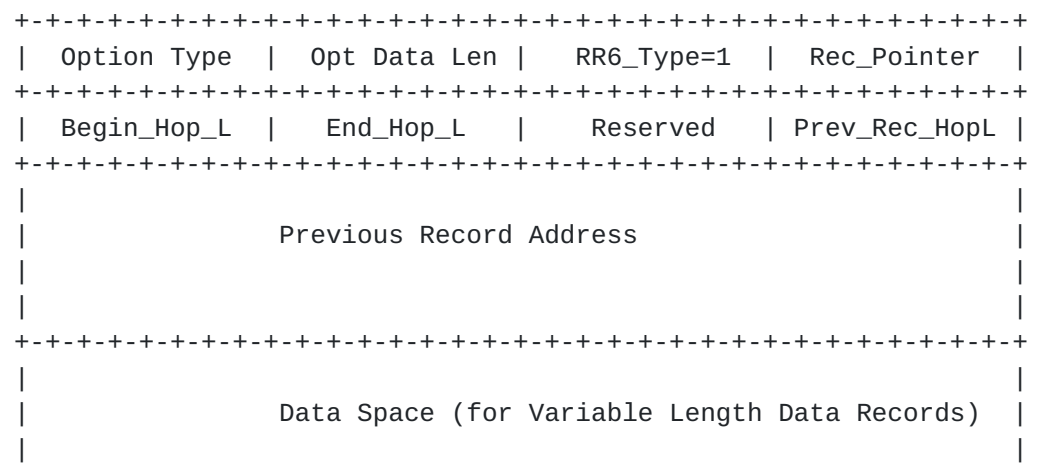
2. Find the same leading part between them in octet

3. Omit the same leading part in octet, and calculate the length of the remaining part of the current address information.

3.3.2 Data Space Format Extension for Compressed Address Record Type

In order to compare current and previous information, the leading 17 octets of the [Data Space] are used to keep Hop Limit and Address information of the previous (last data recorded) node.

The following format is used in the compressed address record type. (This format is an upper compatible format from the basic RR6 format)



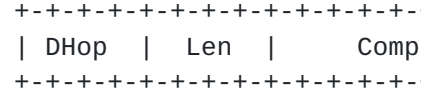
Prev_Rec_HopL: 8-bit unsigned integer.
Hop Limit value of the previous (last data recorded) node

Previous Record Address: 16-octet
(Incoming interface's) address of the previous (last data recorded) node

(If [Begin_Hop_L] >= Hop Limit value of the source node, the source node fills [Prev_Rec_HopL] with Hop Limit value of the source node, fills [Previous Record Address] with outgoing interface's address information of the source node, and sets [Rec_Pointer] with 17)

3.3.3 Variable Length Data Record Format

The following is a format of one variable-length data record.



(Maximum length: 17 octets)

DHop: 4bit.
Hop difference that is calculated by the following formula [Prev_Rec_HopL] - [Current Hop Limit]
(=0 means more than 15 hop difference)

Len: 4bit.
Length of the following [Compressed Address]
(=0 means 16)

Compressed Address:
Compressed current address
(The same leading part in octet is omitted)

(Max length 16)

Note:

In usual cases (hop differences are smaller than 16), [DHop] can show accurate hop difference. Only when some hop difference on the path is larger than 15, a situation ([DHop] = 0) appears in the [Data Space], and this causes a Hop Limit counting ambiguity.

If the situation ([DHop] = 0) appears only once in the [Data Space], such an ambiguity can be cleared by checking the initial Hop Limit value and the Hop Limit value of the received packet.

If the situation ([DHop] = 0) appears more than once in the [Data Space], such ambiguities are not cleared in the compressed address record type. If it is necessary to count Hop Limit values of nodes accurately, the plain (non-compressed) address record type accompanied with appropriate data recording range control is used.

3.3.4 Operation Steps at Each Node

1. Prepare a variable-length data record
   Calculate "DHop", "Compressed Address" and "Len"

2. Write the data record into [Data Space (for Variable Length Data Records)]. Writing start point in [Data Space] is indicated by [Rec_Pointer]

3. Update [Rec_Pointer] ([Rec_Pointer] += (1+[Len]))

4. Update [Prev_Rec_HopL] with Hop Limit value of current node

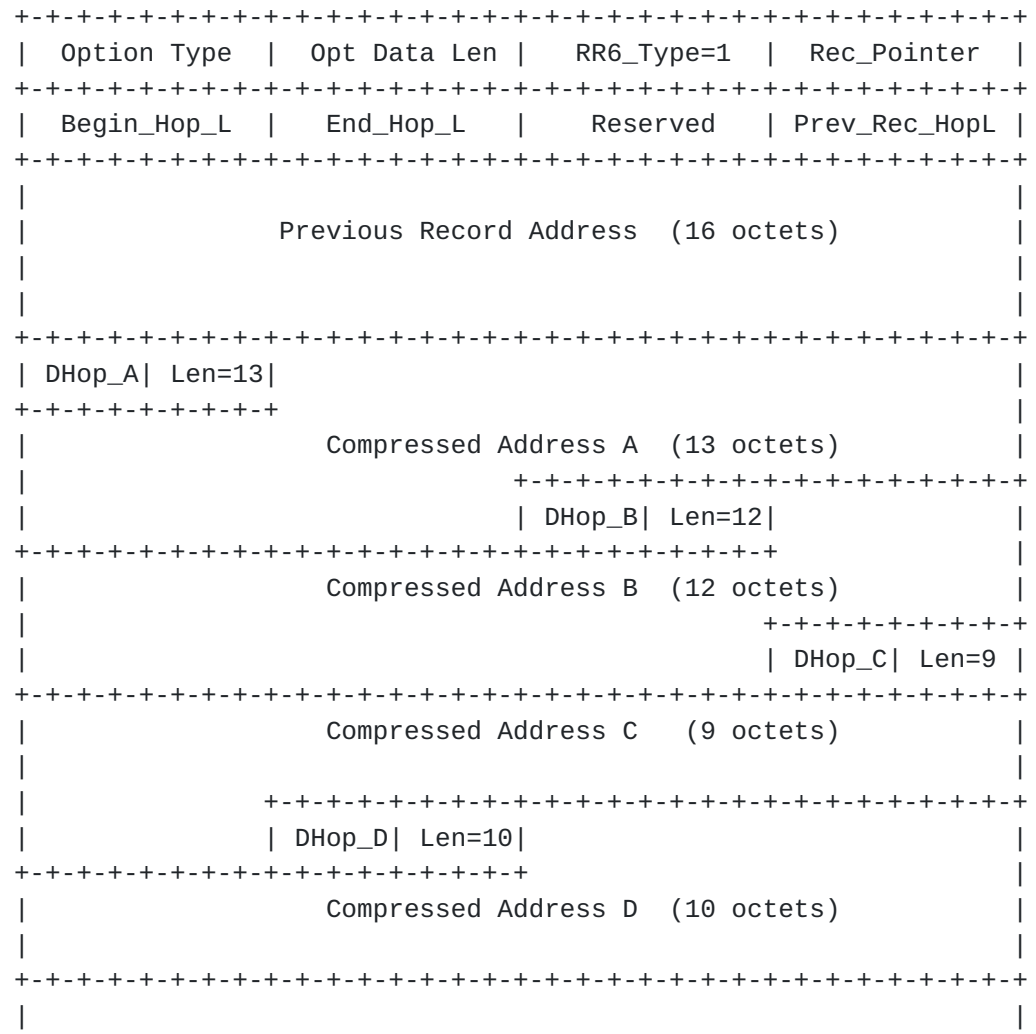
5. Update [Previous Record Address] with address information of current node

Note:

Only step 4 and 5 are executed on an initial data record node, and no data is recoded into [Data Space (for Variable Length Data Records)].

3.3.5 Example of Data-Filled Format (Compressed Address Record Type)

The followings show an example how the [Data Space] is filled with variable-length data records.



4. IANA Considerations

The RR6 mechanism has introduced one IPv6 Hop-by-Hop option. In order to implement the RR6 mechanism, one IPv6 Hop-by-Hop option type value is tentatively assigned from unassigned spaces as follows.

IPv6 Hop-by-Hop Option: RR6 Type: 0x23 (Tentatively)

This number must be assigned by IANA officially.

5. Security Considerations

Since address information is not secret information of nodes, the Record Route for IPv6 (RR6) mechanism not have any direct impact on Internet infrastructure security.

Appendix A. Recorded Hop Limit value of the Node

A node deals with two Hop Limit values on it. One is set to receiving packets; the other is set to transmitting packets. The timing when Hop Limit value is changed (decremented) on the node is not clearly defined. It is necessary that which values is appropriate to be recorded by the RR6 option operations.

Since the recorded Hop Limit value of the node is used to identify its location on the communication path, it must be unique and different from Hop Limit values of other nodes on the path.

The initial Hop Limit value is not decremented on the source node, so that the initial Hop Limit value is adopted for the recorded Hop Limit value of the source node. Thus, the first node (succeeding node from the source node on the path) can not use the initial Hop Limit value, and the decremented initial Hop Limit value is adopted for the recorded Hop Limit value of the first node.

Therefore, the rule is inductively defined. The recorded Hop Limit value of the node is the Hop Limit value that is set to transmitting (outgoing) packets. In other words, the decremented Hop Limit value that is set to receiving (incoming) packets is used for the recorded Hop Limit value of the node.

Appendix B. RR6 Implementation

The main RR6 operations are implemented at incoming operations (e.g., ip6_input()). Some RR6 operations, however, are implemented at outgoing operations (e.g., ip6_output()), because the RR6 mechanism must run on the source node.

References

[IPv6] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," RFC2460, December 1998.

[ICMPv6] A. Conta, S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," RFC2463, December 1998.

Author's Address:

Hiroshi Kitamura
NEC Corporation
Development Laboratories
(Igarashi Building 4F) 11-5, Shibaura 2-Chome,
Minato-Ku, Tokyo 108-8557, JAPAN

Phone: +81 (3) 5476-1071
Fax: +81 (3) 5476-1005
Email: kitamura@da.jp.nec.com